INFORMATION AND CONTROL **43**, 198–201 (1979)

A Remark on Tally Languages and Complexity Classes*

RONALD V. BOOK

Department of Mathematics, University of California at Santa Barbara, Santa Barbara, California 93106

A condition on a class of languages is developed. This condition is such that every tally language in that class is accepted in polynomial time by a deterministic Turing machine if and only if every language accepted in exponential time by a nondeterministic machine is also accepted in exponential time by a deterministic machine.

A tally language is a language over a one-letter alphabet. In a previous paper (Book, 1974) it was shown that the following are equivalent:

(i) Every tally language accepted in real time by a nondeterministic Turing machine is accepted in polynomial time by a deterministic Turing machine.

(ii) Every language accepted in exponential time by a nondeterministic Turing machine is also accepted in exponential time by a deterministic Turing machine.

The purpose of this note is to provide a somewhat weaker condition than (i) that also is equivalent to (ii). This condition can be applied to several classes of languages in addition to the class NTIME(n) of languages accepted in real time by nondeterministic machines since it applies to any class with two specific properties, whether the class is specified algebraically or by a class of machines or by a class of grammars. In particular, it can be applied to three subclasses of NTIME(n).

Let Σ be a finite set of symbols. If Σ contains just k symbols, then identify Σ with the set of digits $\{1, 2,..., k\}$ and identify each $w \in \Sigma^*$ with an integer $n(w)$ in k-adic notation.

For a string w, $|w|$ is the length of w.

If M is a Turing machine, then $L(M)$ is the set of strings accepted by M. For a function T of the length of the input, DTIME(T) $= \{L(M) \mid M$ is a deterministic machine that runs in time $T\}$ and NTIME(T) $= \{L(M) \mid M$ is a nondeterministic machine that runs in time $T\}$. Let $P = \bigcup_{j\geq 1} \mathrm{DTIME}(n^j)$,

* This research was supported in part by the National Science Foundation under Grant MCS77-11360.

0019-9958/79/110198–04$02.00/0

$NP = \bigcup_{j\geq 1} \mathrm{NTIME}(n^j)$, $\mathrm{DTIME}(2^{\mathrm{lin}}) = \bigcup_{c>0} \mathrm{DTIME}(2^{cn})$, and $\mathrm{NTIME}(2^{\mathrm{lin}}) = \bigcup_{c>0} \mathrm{NTIME}(2^{cn})$.

To state the result, one new notion must be defined. Informally, it describes the idea of representing each language from a class $\mathscr{L}_1$ in another class $\mathscr{L}_2$ by allowing polynomial padding.

Let L_1 be a language, let Σ be a finite alphabet such that $L_1 \subseteq \Sigma^*$, and let d be a new symbol, $d \notin \Sigma$. Suppose that $t > 0$ is an integer. Then the language $L_2 = \{d^q w \mid w \in L_1, q = |w|^t\}$ is a *strict polynomial representative* of L_1.

THEOREM. *Let $\mathscr{L}$ be a class of languages with the following properties:*

(i) *Every tally language in NP has a strict polynomial representative in $\mathscr{L}$.*

(ii) *$\mathscr{L}$ is closed under nonerasing homomorphism.*

(iii) *Every tally language in $\mathscr{L}$ is in NP.*

Every tally language in $\mathscr{L}$ is accepted by a deterministic polynomial time-bounded Turing machine if and only if every language accepted by a nondeterministic Turing machine which operates within time bound 2^{cn} for some $c > 0$ is also accepted by a deterministic Turing machine that operates within time bound 2^{dn} for some $d > 0$, that is, $\{L \mid L \subseteq \{1\}^*, L \in \mathscr{L}\} \subset P$, if and only if $\mathrm{DTIME}(2^{\mathrm{lin}}) = \mathrm{NTIME}(2^{\mathrm{lin}})$.

Proof. First assume that $\{L \mid L \subseteq \{1\}^*, L \in \mathscr{L}\} \subset P$. Let $L_1 \in \mathrm{NTIME}(2^{\mathrm{lin}})$. Let Σ be a finite alphabet such that $L_1 \subseteq \Sigma^*$ and let k be the number of symbols in Σ. Let L_2 be the tally language corresponding to L_1, i.e., $L_2 = \{1^{n(w)} \mid w \in L_1\}$. Now L_2 is in NP (Book, 1974, Lemma 2) so for some $q \geq 1$, $L_2 \in \mathrm{NTIME}(n^q)$. Since L_2 is a tally language in NP and by hypothesis every tally language in NP has a strict polynomial representative in $\mathscr{L}$, there is a symbol $d \neq 1$, an integer $t > 0$, and a language L_3 in $\mathscr{L}$ such that $L_3 = \{d^p y \mid y \in L_2, p = |y|^{t+1}\}$. Let $h: \{1, d\}^* \to \{1\}^*$ be the homomorphism determined by defining $h(1) = h(d) = 1$, and let $L_4 = \{h(y) \mid y \in L_3\}$. By hypothesis $\mathscr{L}$ is closed under non-erasing homomorphism so L_4 is in $\mathscr{L}$.

Since L_4 is a tally language in $\mathscr{L}$, L_4 is in P so that there is a deterministic Turing machine M_4 and a constant r such that $L(M_4) = L_4$ and M_4 runs in time n^r. From M_4 we construct a machine M_1 for L_1.

M_1 has input alphabet Σ. On input $w \in \Sigma^*$, M_1 writes the integer $n(w)$ corresponding to w in tally notation on one of its work tapes. This process takes at most $2 \cdot k^{|w|}$ steps. Then M_1 writes $1^{n(w)+p}$, where $p = n(w)^t$, on another work tape. Finally, M_1 simulates M_4 on $1^{n(w)+p}$. If M_4 accepts $1^{n(w)+p}$, then M_1 accepts w; otherwise, M_4 rejects w.

Now M_1's computation on w has at most $2 \cdot k^{|w|} + n(w)^{t+1} + (n(w)^{t+1})^r$ steps. Since $n(w) \leq k^w$, there is an integer s such that M_1's computation on w has at most $2^{s|w|}$ steps. Clearly M_1 is deterministic so $L(M_1) \in \mathrm{DTIME}(2^{\mathrm{lin}})$.

To see that $L_1 = L(M_1)$, notice that the mapping $w \mapsto 1^{n(w)}$, $w \in \Sigma^*$, is one-to-one, as is the mapping $1^{n(w)} \mapsto d^p 1^{n(w)}$, $p = n(w)^t$. Thus the homomorphism h is one-to-one on L_3. Hence, $L_1 = L(M_1)$. Since L_1 was chosen arbitrarily from NTIME(2^{lin}), we see that DTIME(2^{lin}) = NTIME(2^{lin}).

Now assume that DTIME(2^{lin}) = NTIME(2^{lin}). Note that DTIME(2^{lin}) = NTIME(2^{lin}) implies that $\{L \mid L \subseteq \{1\}^*, L \in NP\} \subset P$ and that for any language L and any polynomial g, if $\{1^{(g|w|)} \mid w \in L\}$ is in P (NP), then so is the language $\{1^{|w|} \mid w \in L\}$. Since $\{L \mid L \subseteq \{1\}^*, L \in \mathscr{L}\} \subset NP$, this means that every tally language in $\mathscr{L}$ is also in P. ∎

Three examples of subclasses of NTIME(n) meeting the conditions for the class $\mathscr{L}$ of the theorem are the following.

1. The class $\mathscr{L}_{\text{BNP}}$ is the class of languages accepted in linear time by nondeterministic Turing machines whose read–write heads are reversal-bounded. Equivalently, the class $\mathscr{L}_{\text{BNP}}$ is the smallest class of languages containing $\{ww^R \mid w \in \{a, b\}^*\}$ and closed under intersection and nonerasing homomorphism (Book, Nivat, and Paterson, 1974).

2. The class LINEAR$_{\text{CS}}$ is the class of languages generated by context-sensitive grammars having linear derivational complexity, i.e., linear time bounds on derivations (Book, 1971, 1978).

3. The class MULTI-RESET is the class of languages accepted in linear time by nondeterministic Turing machines whose work tapes are single reset tapes. Equivalently, the class MULTI-RESET is the smallest class of languages containing $\{ww \mid w \in \{a, b\}^*\}$ and closed under intersection and nonerasing homomorphism (Book, Greibach, and Wrathall, 1979).

It is conjectured in Book, Greibach, and Wrathall (1979) that MULTI-RESET $\subsetneq \mathscr{L}_{\text{BNP}} \subsetneq$ NTIME(n) and it is known that $\mathscr{L}_{\text{BNP}}$ = NTIME(n) (resp., MULTI-RESET = NTIME(n)) if and only if every context-free language is in $\mathscr{L}_{\text{BNP}}$ (resp., MULTI-RESET). It is known that every context-free language is in LINEAR$_{\text{CS}}$ and that $\{ww \mid w \in \{a, b\}^*\}$ is not in LINEAR$_{\text{CS}}$ (Book, 1971) but is in MULTI-RESET $\subseteq \mathscr{L}_{\text{BNP}}$, so that LINEAR$_{\text{CS}}$ is not comparable to either MULTI-RESET or $\mathscr{L}_{\text{BNP}}$ unless the appropriate class contains every context-free language. Each of the classes $\mathscr{L}_{\text{BNP}}$, MULTI-RESET, and LINEAR$_{\text{CS}}$ have NP-complete sets so that if one of these classes is included in P, then $P = NP$ and DTIME(2^{lin}) = NTIME(2^{lin}).

The theorem presented here gives conditions on a class $\mathscr{L}$ of languages such that $\{L \mid L \subseteq \{1\}^*, L \in \mathscr{L}\} \subset P$ if and only if DTIME(2^{lin}) = NTIME(2^{lin}). One would like to find minimal conditions on $\mathscr{L}$ for this equivalence to hold, or to find minimal classes $\mathscr{L}$ such that this equivalence holds. We conjecture that both MULTI-RESET and LINEAR$_{\text{CS}}$ are minimal classes.

The technique used in the proof of the theorem has been used by Monien (1979) in his study of the class of languages accepted by Turing machines that use linear work space.

RECEIVED: January 2, 1979; REVISED: May 24, 1979

References

BOOK, R. (1971), Time-bounded grammars and their languages, *J. Comput. System Sci.* **5**, 397–429.

BOOK, R. (1974), Tally languages and complexity classes, *Inform. Contr.* **26**, 186–193.

BOOK, R. (1978), On the complexity of formal grammars, *Acta Inform.* **9**, 171–181.

BOOK, R., GREIBACH, S., AND WRATHALL, C. (1979), Reset machines, *J. Comput. System Sci.* **19** (1979), in press.

BOOK, R., NIVAT, M., AND PATERSON, S. (1974), Reversal-bounded acceptors and intersections of linear languages, *SIAM J. Comput.* **3**, 283–295.

MONIEN, B. (1979), The LBA-problem and the deterministic tape complexity of two-way one-counter languages over a one-letter alphabet, unpublished manuscript.